A Service Flow Management Strategy for IEEE 802.16 Broadband Wireless Access Systems in TDD Mode

Jianfeng Chen, Wenhua Jiao, Hongxi Wang
Lucent Technologies, Bell Labs Research China,
{chenjf, wjiao}@lucent.com

Abstract- ***A*** **fair and efficient service flow management architecture for IEEE802.16 Broadband Wireless Access (BWA) systems is proposed for TDD mode. Comparing with the traditional fixed bandwidth allocation, the proposed architecture adjusts uplink and downlink bandwidth dynamically to achieve higher throughput for unbalanced traffic. A deficit fair priority queue scheduling algorithm is deployed to serve different types of service flows in both uplink and downlink, which provides more fairness to the system. Simulation results show the proposed architecture can meet the QoS requirement in terms of bandwidth and fairness for all types of traffic.**

Key Words: IEEE 802.16, BWA, scheduling, QoS

I. I. Introduction

IEEE published IEEE 802.16[1-2] standard to provide high-speed wireless access in Metropolitan Area Network in 2004. For point-to-multipoint (PMP) topology, a controlling base station (BS) connects multiple subscriber stations (SS) to various public networks. The standard defines a connection-oriented MAC protocol, and a mechanism for QoS guarantee. However, scheduling algorithms for uplink and downlink bandwidth allocation in a single frame are left undefined.

TDD mode has advantage over FDD in that the subframe length of uplink and downlink may be flexibly allocated to serve the un-balanced traffic, which is the only duplex mode considered in this paper. Several approaches for bandwidth allocation for TDD mode in BWA system can be found in [3-4]. However, they only consider the scheduling for uplink sub-frame, and a common assumption is that uplink and downlink sub-frame cover fixed proportion bandwidth each. For example, in [4], the proportion is 50% for uplink and 50% for downlink in its simulation. Obviously this scenario does not happen frequently in the realistic transmission.

To the best of our knowledge, there is no proposed bandwidth allocation solution considering uplink and downlink simultaneously. In this paper, first, an admission control policy is defined to control the number of service flow. Then bandwidth allocation architecture is proposed to provide QoS support for different types of applications. By serving uplink and downlink services dynamically, bandwidth can be utilized more efficiently. Since most paper [3-4] applies strict Priority Queue (PQ) for different class of service, which leads starvation of low priority service when the higher priority service is heavy, the proposed DFPQ will improve the fairness over the previous research.

The paper is organized as follows. In section II, new service flow management strategy is proposed. Section III provides simulation results of the proposed algorithm. Section IV concludes the paper.

II. Proposed service flow management for IEEE 802.16

A. The hierarchical structure of the bandwidth allocation

The QoS management for IEEE 802.16 has the following components: (1) Admission control. It is used to limit the number of flows admitted into the network so that overflow and starvation for some services can be restricted. (2) Buffer management. It is deployed to control the buffer size and decide which packets to drop. (3) Scheduling. It is adopted to determine which packet will be served first in a specific queue to guarantee its QoS requirement.

Since IEEE802.16 MAC protocol is connection oriented, the application must establish the connection with the BS as well as the associated service flow (UGS, rtPS, nrtPS or BE). BS will assign the connection with a unique connection ID (CID) to each uplink or downlink transmission. As shown in Fig.1, when a new service flow generates or updates its parameters, it will send message (DSA/DSC) to the BS. Then the admission control in BS will determine whether this request will be approved or not. Bandwidth request will be sent out for each flow. All bandwidth requests from the services are classified by the connection classifier based on CID and its service type, they are forwarded to the appropriate queue. The scheduling module will retrieve the requests from the queues and generate UL-MAP and DL-MAP message according to the bandwidth allocation results.

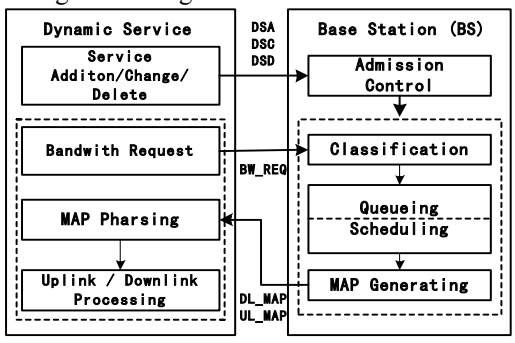

Fig.1. Module diagram of BS and SS

To support all types of service flows, a hierarchical scheduling structure of the bandwidth allocation is proposed for TDD mode. The scheduling uses a combination of Deficit Fair Priority Queue (DFPQ) for multiple service flow, round robin for BE[5], earliest deadline first (EDF) for rtPS[6] and weight fair queue (WFQ) for nrtPS[7].

The hierarchical structure of the bandwidth allocation is shown in Fig. 2. In this architecture, two-layer scheduling is deployed. Six queues are defined according to their direction (uplink or downlink) and service classes. Since UGS will be allocated fixed bandwidth (or time duration) in transmission, their bandwidths will be directly cut before each scheduling.

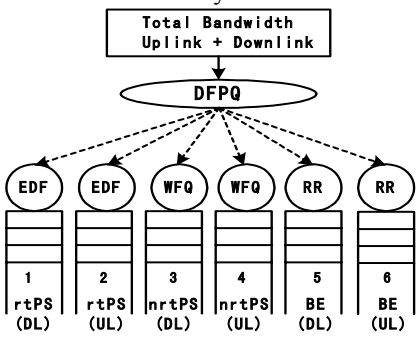

Fig.2. Hierarchical structure of bandwidth allocation

Bandwidth requirement can be measured by the maximum sustained traffic rate (r_{max}) and the minimum reserved traffic rate (r_{min}). It is carried in the DSA and DSC message at the beginning period of connection setup. They are used to negotiate their traffic parameters. In this paper, the minimum reserved traffic rate is used for admission control. The maximum sustained traffic rate is used for scheduling.

For rtPS service, both r_{max} and r_{min} should be specified (not equal to zero) in DSA/DSC message. For nrtPS service, r_{min} should be specified (not equal to zero) in DSA/DSC message at least. For BE service, neither r_{min} nor r_{max} can be specified. If r_{min} is not specified, we suppose that the user allows the provider with a higher priority to occupy all his bandwidth.

B. Admission control

The admission control mechanism determines whether a new request for a connection can be granted or not according to the remaining free bandwidth. . As described in [1], there are 4 types of MAC layer services. These service flows can be created, changed, or deleted through the issue of DSA, DSC, and DSD messages. Each of these actions can be initiated by the SS or the BS and are carried out through a two or three-way-handshake.

BS should have an admission control policy to decide whether QoS of a connection can be satisfied. One principle is to ensure the existing connection's QoS will not be degraded significantly and the new connection's QoS will be satisfied, which is adopted as the QoS policy in this paper.

A simple admission control is adopted by using the Minimum Reserved traffic rate. It will collect all the DSA/DSC/DSD requests and update the estimated available bandwidth (C_a) based on bandwidth change. Suppose there are I classes of service and the i^{th} classes of service has totally J_i connection, the available bandwidth equals to:

$$C_a = C_{tatal} - \sum_{i=0}^{I} \sum_{j=0}^{J_i - 1} r_{\min}(i,j) \qquad (1)$$

In which $r_{min}(i,j)$ is the Minimum Reserved traffic rate of the j^{th} connection in the i^{th} class of service flow, C_{total} is the total capacity of the wireless link. For those connections whose r_{min} is equal to zero, they can always be accepted by our admission policy. But the QoS of these connections will not be guaranteed. They always have the lowest priority. Their connections will be interrupted anytime unless the QoS requirements of all other connections can meet sufficiently.

When a new service flow comes or an old service flow requests to change its QoS, the following principle should hold:

$$C_a \geq 0 \qquad (2)$$

In this paper, (2) is the QoS policy for admission control.

C. Scheduling architecture

1) First layer scheduling: Deficit Fair Priority Queue (DFPQ)

In [4], the strict priority queue is used, in which bandwidth allocation per flow follows strict priority, from highest to lowest: UGS, rtPS, nrtPS and BE. Similar to [4], DFPQ is basically based on priority queue. The initial priority is defined as the following 2 policies:

1. Service class based priority: *rtPS > nrtPS> BE*
2. Transmission direction based priority: *Downlink > Uplink*

The reason why the priority of Downlink is higher than uplink is that in a central scheduling architecture, the BS needs to relay packets as soon as possible to avoid buffer overflow and guarantee latency requirements. Based on this, the priority shown in Table 1 is used in this paper.

Table I. Priority of each service class

DL-rtPS	UL-rtPS	DL-nrtPS	UL-nrtPS	DL-BE	UL-BE
1	2	3	4	5	6

One disadvantage of the strict priority service is that higher priority connections can starve the bandwidth of lower priority connections. To overcome this problem, DFPQ algorithm is proposed in this paper to improve the fairness.

There is an active list maintained in BS. The DFPQ only schedules the bandwidth application services in active list. If the queue is not empty, it will stay in active list. Otherwise, it will be removed from active list. The service flows in the active list are queued by strict priority shown in Table I. In each round, highest priority flow will always be served first.

DFPQ is proposed referring to DWRR. DWRR scheduling was proposed in 1995 [8]. In the classic DWRR algorithm, the scheduler visits each non-empty queue and determines the number of bits in the packet at the head of the queue. The variable *DeficitCounter* is incremented by the value quantum. Similar to [8], the variable *DeficitCounter* is defined in DFPQ. In this paper, bandwidth requests will be serviced in the queue. The scheduler visits each non-empty queue in the

active list and determines the number of request in this queue. The variable *DeficitCounter* is incremented by the value *Quantum* each time when it is visited. If the requested data size of the BW-REQ packet at the head of the queue is less than or equal to the variable *DeficitCounter*, the variable *DeficitCounter* is reduced by the number of bits in the packet and the packet is transmitted to the output port. The process will be repeated until either the *DeficitCounter* is no more than zero or the queue is empty. If the queue is empty, the value of *DeficitCounter* is set to zero too. When this condition occurs, the scheduler moves on to serve the next non-empty priority queue.

The next problem is how to determine the *Quantum*. The solution is that the *Quantum[i]* for the i^{th} class of service flow is decided by

$$Quantum[i] = \sum_{j=0}^{J_i} r_{\max}(i,j) \qquad (3)$$

In which the J_i is the total connections for the ith class of service flow. Notes that for the connection whose r_{max} is not specified (e.g. r_{max} = 0 in DSA message), r_{max} will be set to r_{min} in (3). The service rate of the connection that has the larger *Quantum* will be higher than the smaller one.

The DFPQ scheduling algorithm is illustrated by an example shown in Fig. 3, in which L_a is available capacity of a TDD frame in bits, L_{total} is total capacity of a TDD frame.

Step 1: Update the active list.

If a connection's waiting queue is empty, remove it from active list. If a connection's waiting queue has changed from empty to being occupied, add it to the active list. The service flows in the active list will be queued by priority in Table 1.

Step 2: Update the parameters for each service queue.

The initial value of L_a is: $L_a = L_{total.}$

The initial value of *DeficitCounter[i]* is:

DeficitCounter [i] = Quantum[i].

Step 3: Serve the connections in the service flow with the first priority queue one by one until one of the conditions is satisfied:

Cond. (1) *DeficitCounter [i]*≤0.

Cond. (2) BW-REQ waiting queue is empty.

Cond. (3) No available bandwidth left, namely, L_a ≤0.

Cond. (4) It is time sending MAP message.

In this example, at first, because the 600-bit BW request (BW-REQ) packet at the head of rtPS queue is smaller than the value of *DeficitCounter[rtPS]=1000*, the 600-bit BW-REQ is serviced. This cause the *DeficitCounter[rtPS]* to be decremented by 600 bits, resulting in a new value of 400. Now, since the 300-bit BW-REQ at the head of rtPS queue is smaller than 400, the 300-bit packet is also being serviced, creating a new value of 100. The BW-REQ in rtPS queue will be serviced until the *DeficitCounter[rtPS]*=-300 (<0).

Step 4: If Cond. (1) or Cond. (2) stands, service other lower priority queues as in Step 3. If there is no lower priority queue, go to Step 3 for anther scheduling round.

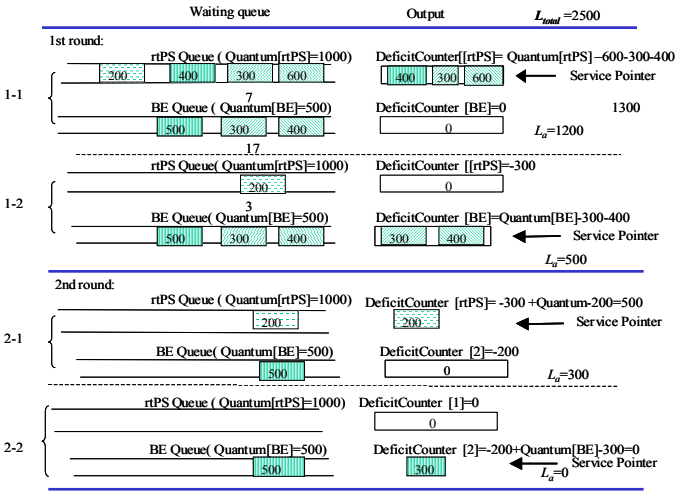

Fig.3. An example of DFPQ scheduling

In this example, BE queue will be serviced until *DeficitCounter [BE]* <0. Since there is no other service flow and there still some available bandwidth, it goes to the next round.

In the 2nd round, rtPS flow is the highest priority. At first, *DeficitCounter[rtPS]* will be the sum of *Quantum[rtPS]* and *DeficitCounter[rtPS]* in previous round, that 700. Then the 200-bit BW request will be processed. After that, *DeficitCounter[rtPS]* = 500.

Step 5: If Cond. (3) or (4) stands, send the MAP message out and end the scheduling for current TDD frame. Go to *Step 1* for next TDD frame.

In this example, Cond. (3) is satisfied at the end of the 2nd round (*Step 2*). So, the scheduling ends for this TDD frame. Notes that, in step 2 of round 2, there are only 300 bits acknowledged in BS for 500-bit BW-REQ. The SS should do fragmentation for the 500-bit packet. The first 300-bit sub-frame should be transmitted, and the left 200-bit sub-frame should apply for transmission again in the next TDD frame. From above, in each service round, the highest priority queue will be served first until its assigned bandwidth is deficit. So, this is a priority queue based scheduling. On the other hand, if the assigned bandwidth is deficit for the higher priority queue, the service flow with lower priority queue will have chance to be served, which seems a fair solution for the lower priority queue and differs from the strict priority queue. That is why the proposed scheduling algorithm is called Deficit Fair Priority Queue.

2)Second layer scheduling

Three different algorithms are assigned to three classes of service to match its requirements.

rtPS connections: EDF. Packets with earliest deadline will be scheduled first. The information module determines the packets' deadline and the deadline are calculated by its arrival time and maximum latency.

nrtPS connections: WFQ. We schedule nrtPS packets based on the weight (ratio between a connection's nrtPS Minimum Reserved Traffic Rate and the total sum of the Minimum Reserved Traffic Rate of all nrtPS connections).

BE connections: The remaining bandwidth is allocated to each BE connection by round robin (RR).

The detailed scheduling algorithm for EDF, WFQ and RR can be seen from reference.

3)Buffer management

Buffer management is used to control the buffer size and decide which packets to drop. Timing sensitive traffic has its maximum delay requirement. Buffer management will drop those packets that exceed their maximum latency, which directly contributes for packet loss rate. It is supposed the buffer for each service flow is infinite.

III. Simulation results

The assumption of total bandwidth is C_{total} =10 Mbps and the duration for each frame f is 10ms, so the bandwidth for a frame is 100Kbit. The parameters used in the simulation are given in Table II. It is supposed that the r_{max} and r_{min} fluctuates about ±20 percent over the average traffic rate. All packet arrivals occur at the beginning of each frame and the packet arrival process for each connection follows the Poisson distribution with different traffic rate λ. Each connection has specific QoS parameters in terms of Maximum Sustained Traffic Rate, Minimum Reserved Traffic Rate and Maximum Latency requirement. For all types of service flow, *Quantum* value can be calculated in (3).

Table II. Input Traffic

CID	Type	Average Bandwidth (Kbit)	Quantum (Kbit)	Maximum Delay (ms)	Max.sustained rate (Kbit)	Min. reserved rate (Kbit)
1	DL_rtPS	10	36	60	12	8
2		10		40	12	8
3		10		20	12	8
4	UL_rtPS	7	24	70	8.4	5.6
5		7		50	8.4	5.6
6		6		30	7.2	4.8
7	DL_nrtPS	6	21.6	120	7.2	4.8
8		6		80	7.2	4.8
9		6		60	7.2	4.8
10	UL_nrtPS	4	14.4	140	4.8	3.2
11		4		150	4.8	3.2
12		4		60	4.8	3.2
13	DL_BE	2	4.8	240	-	1.6
14		2		200	-	1.6
15		2		160	-	1.6
16	UL_BE	2	3.2	280	-	1.6
17		1		240	-	0.8
18		1		200	-	0.8

Throughput and traffic rate are investigated in performance study. Suppose there are total N frames scheduled by the system for the service i, throughput Th_i (bit) is calculated by:

$$Th_i = \sum_{j=1}^{N} map_{ij} \quad (4)$$

In which map_{ij} is the served data size of service i in frame j. The traffic rate Tr_i(bps) is as follows:

$$Tr_i = Th_i / t_d \quad (5)$$

In which t_d is the scheduling duration for N frames.

Fig. 4-5 show the arrival and service curves of nrtPS and BE services. We compare the throughput performance of dynamic bandwidth assignment strategy (general scheduling architecture for both uplink and downlink traffic shown in Fig. 2) and the fixed one (50% for uplink, 50% for downlink). Since in most cases, the downlink traffic has larger proportion, it is assumed the proportion of input downlink and uplink traffic is 6:4 in Fig. 4-5.

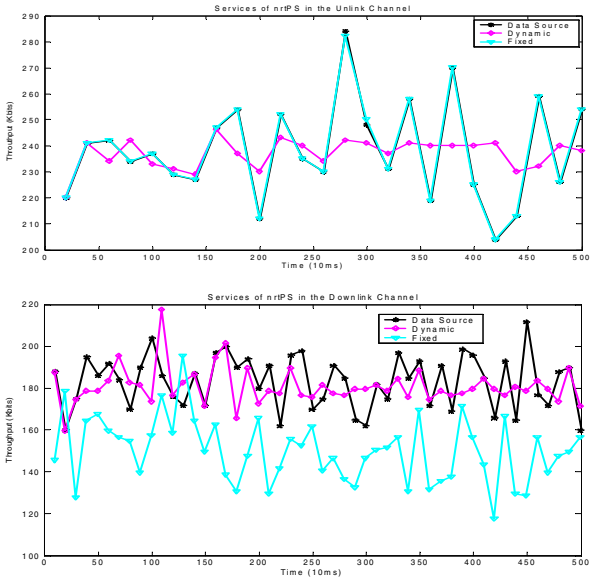

Fig. 4 Throughput of nrtPS services

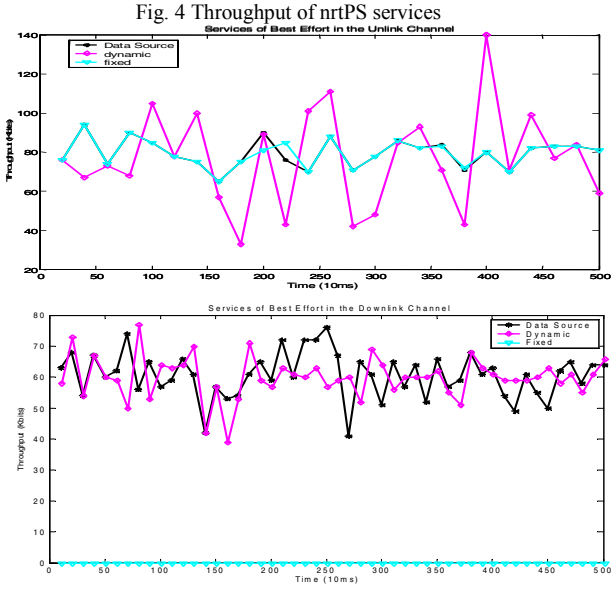

Fig. 5. Throughput of BE services

In the uplink channel, in general, both service curves adapt and follow the arrival curve (actually, the fixed one is better because there is enough bandwidth for the uplink traffic).

However, the proposed scheduling architecture outperforms in downlink channel, which is the contribution of dynamically allocation of bandwidth both in uplink and downlink based on the demand of each session. For fixed bandwidth allocation, although uplink flows are served well, services rate of nrtPS and BE in downlink channel can not meet their minimum reserved data rate because the downlink data exceeds the fixed bandwidth boundary of 5Mbits. The advantage of the dynamically bandwidth assignment strategy is that it will automatically balance the uplink and downlink traffic.

We also compare the performance of the two first-layer-scheduling algorithms (Priority Queue vs. proposed DFPQ) using the same admission control and buffer management described in section II. As shown in Fig. 6, the scenario of the top graph is the bandwidth of rtPS services rising from 40Kbit/frame to 70Kbit/frame (result in the total bandwidth rising from 90Kbit/frame to 120Kbit/frame), while other service flow unchanged. The conclusion is under PQ scheduling rtPS services are always served first, so the services of BE are staved when the total incoming traffic arrives 110Kbits/frame (11Mbps, larger than C_{total} 10Mbps); In DFPQ scheduling, the bandwidth for BE decreases to its minimum reserved rate and the service of rtPS is allocated bandwidth up to its Maximum sustained traffic rate. The scenario on the bottom graph is bandwith of nrtPS varies from 20Kbits/frmae to 50Kbits/frame (result in the total bandwidth rising from 90Kbit/frame to 120Kbit/frame), while other service flow unchanged. Two scheduling methods can both serve the flows of rtPS well, the situation of BE traffic is quite the same as the previous scenario in PQ scheduling: This kind of service begins to starve when the total incoming traffic arrives more than the capacity. But for DFPQ, the minimum reserved traffic rate can still be guaranteed.

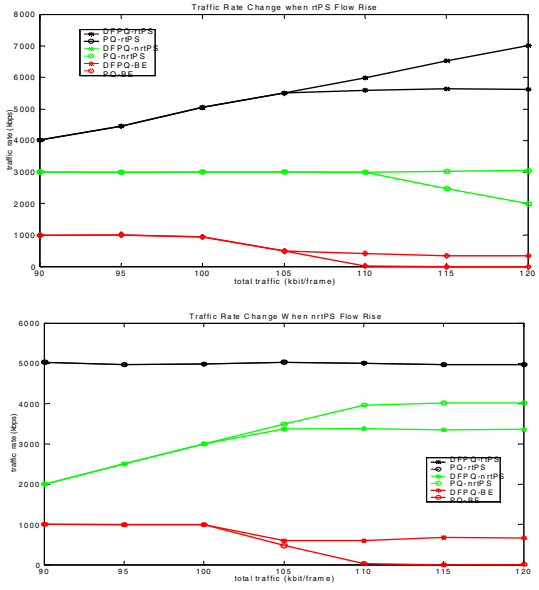

Fig..6. Served traffic rate changes when rtPS and nrtPS service flows rise

A scheduling algorithm is said to be fair if the difference in normalized services received by different flows in the scheduler is bounded. Let S_{rtps} be the total traffic source of rtPS service and S_{be} be the total traffic of BE service, we define the formula of fairness between rtPS and BE service ($FAIR_{r\_b}$) as following:

$$FAIR_{rb} = \left| \frac{Th_{rtps}}{S_{rtps}} - \frac{Th_{be}}{S_{be}} \right| \quad (6)$$

Fig. 7 shows the fairness between rtPS and BE when a rtPS service flow rises. In Fig. 7, when total traffic under 100Kbit/frame, two scheduling methods have same fairness boundary. When the total traffic exceeds 100Kbit/frame, DFPQ algorithm can still keep the fairness boundary under 0.1, however, because all services of BE are starved, the curve shows PQ scheduling is quite unfair.

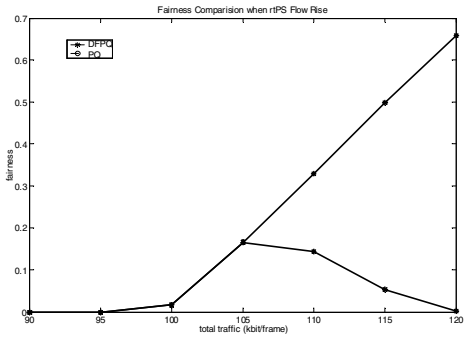

Fig. 7. Fairness comparison

IV. Conclusion

A 2-layer service flow management architecture for IEEE 802.16 standard (TDD mode) is proposed in this paper. Compared with fixed bandwidth allocation, the proposed solution improves the performance of throughput under unbalanced uplink and downlink traffic. What's more, better performance in fairness can be achieved by the proposed DFPQ algorithm than strict PQ scheduling.

Reference

[1]. IEEE 802.16 Standard-Local and Metropolitan Area Networks-Part 16. IEEE 802.16-2004

[2]. IEEE 802.16 Working Group on Broadband Wireless Access. http://wirelessman.org

[3] Hawa, M.; Petr, D.W.,Quality of service scheduling in cable and broadband wireless access systems, Tenth IEEE International Workshop on Quality of Service, p247-255, 2002

[4] K. Wongthavarawat, and A. Ganz, "Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems", International Journal of Communication Systems, Vol. 16, p81-96, 2003

[5] E. L. Hahne, R. G. Gallager, "Round Robin Scheduling for Fair Flow Control in Data Communication Networks," International Conference on Communications, pp. 103–107, June 1986.

[6] Georgiadis L, etc.Optimal Multiplexing on a Single Link: Delay and Buffer Requirements. IEEE INFOCOM 94; vol. 2, 1994; 524–532.

[7] Demers A, Keshav S, Shenker S. "Analysis and Simulation of a Fair Queuing Algorithm". SIGCOMM CCR 19 1989; 4.

[8] Shreedhar, M. and Varghese, G. "Efficient Fair Queueing Using Deficit Round Robin", Proc. ACM SIGCOMM'95, Vol.25, No.4, pp.231-242, October, 1995